STABLE SINGLE-MACHINE SCHEDULES

Roel Leus* and Willy Herroelen
Katholieke Universiteit Leuven
Leuven, Belgium
Roel.Leus@econ.kuleuven.ac.be; Willy.Herroelen@econ.kuleuven.ac.be

Abstract Robust scheduling aims at the construction of a schedule that is protected against uncertain events. A stable schedule is a robust schedule that changes only little when variations in the input parameters arise. We present a model for single-machine scheduling with stability objective, and we propose a branch-and-bound algorithm for solving an approximative formulation of the model.

Keywords: single-machine scheduling, uncertainty, branch-and-bound, stability

1. Introduction

Manufacturing schedules regularly suffer disruptions from a variety of sources. Despite these uncertainties, a deterministic 'baseline' schedule is usually constructed, which forms the basis for communication with suppliers and subcontractors as well as for commitments to customers [1, 3]. During execution, regular schedule updates will be in order. For coordination purposes, it is desirable that the actual start of each job remain close to its initial baseline starting time. Our goal is to produce a schedule that is protected from the undesirable consequences of rescheduling (and therefore *robust*). A baseline schedule is *stable* if there is little deviation between the baseline and the executed schedule [2].

2. Problem statement

A set of jobs N, $|N| = n$, with deterministic baseline durations d_i is to be scheduled on a single machine. A schedule specifies a starting time s_i for each job i. There is a common deadline ω for all the jobs: $s_i + d_i \leq \omega, \forall i \in N$. The actual duration of i is a stochastic variable

*Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen)

D_i. We assume that job execution does not start before the baseline starting time, which guarantees that actual production will strictly cling to the baseline if no disruptions occur, so $s_i \leq S_i, \forall i \in N$, with S_i the actual starting time of job i. In case of disruption, we right-shift the remaining jobs without re-sequencing. If we define $[i]$ to be the job that is scheduled in the i-th position, then $S_{[1]} = s_{[1]}$ and $S_{[i]} = \max\{s_{[i]}; S_{[i-1]} + D_{[i-1]}\}, i = 2, ..., n$.

A non-negative integer cost c_i is incurred per unit-time deviation in s_i to penalise the resulting system nervousness and shop-coordination difficulties and the delivery delay towards the customer. The expected weighted deviation between *actual* and *planned* job starting times is our stability measure: we minimise $\sum_N c_j E[S_j - s_j]$, with $E[\cdot]$ the expectation operator. Stochastic job duration D_i is discrete. Specifically, we let random variable L_i denote the increase in d_i if i is disrupted, which takes place with probability π_i. Disruption scenarios for job i are gathered in set Ψ_i. Durations for different jobs are independent.

To evaluate the objective-function value, little less can be done in general than to consider all $\prod_N (|\Psi_i| + 1)$ possible combinations of disruption scenarios. We therefore develop a model that focuses only on the main effects of the separate disruption of each of the n jobs. The model assumes that exactly one job suffers a disruption from its baseline duration: value

$$p_i = \frac{\pi_i \prod_{\substack{j=1 \\ j \neq i}}^{n} (1 - \pi_j)}{\sum_{k=1}^{n} \pi_k \prod_{\substack{j=1 \\ j \neq k}}^{n} (1 - \pi_j)}$$

is the probability that job i is the unique disrupted job. The resulting restricted model is useful when disruptions are sparse and spread over time, such that the number of interactions is limited.

In our talk, we present a branch-and-bound algorithm to solve the one-disruption model to optimality. Our computational results show that the output of the model is quite robust to variations in the expected number of disrupted jobs $\sum_N \pi_i$.

References

[1] Aytug, H., M. Lawley, K. McKay, S. Mohan, R. Uzsoy. 2005. Executing production schedules in the face of uncertainties: a review and some future directions. *Eur. J. Oper. Res.* **161** 86–110.

[2] Leus, R. 2003. The generation of stable project plans. Complexity and exact algorithms. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium.

[3] Wu, S., H. Storer, P.-C. Chang. 1993. One-machine rescheduling heuristics with efficiency and stability as criteria. *Comput. Oper. Res.* **20** 1–14.